

FED1 Project Exam 1 Brief

Goal

To put into practice the skills you have learned over the FED1 program. You will demonstrate your ability to interpret a brief, plan your solution, design a user interface, build and test a responsive web application. You will be required to use all of the skills that you have learned this year including design, HTML, CSS and JavaScript.

Learning outcomes

- ☐ I can plan a web application project
- ☐ I can design a UI for a web application
- ☐ I can use HTML, CSS to build a web application interface
- ☐ I can use JavaScript to interact with API data
- ☐ I can use JavaScript to add UI functionality
- ☐ I can test a web application using online tools
- ☐ I can deploy a web application to a static host

Restrictions

- CSS or JS Frameworks are **not** permitted in this project. This includes Tailwind, Bootstrap, Vue, Svelte, React and similar packages.
- The project must make use of HTML, CSS and JavaScript.
- Code borrowed from external sources must be clearly marked with inline comments, including a link to the origin.
- You may use icon and font packs in your project.

Terminology

- User: A public visitor to your site that is not logged in.
- Owner: The logged in manager of the blog.

Brief

You have been hired to build a front-end user interface for an existing API blogging application. The client has asked for a responsive application that allows users to view dynamic blog posts. The client requires admin pages to register, login and manage their blog posts. During testing, you will use your own account that you create to act as the owner.

Client

You may create your own fictional client for this project, with a theme, backstory and branding of your choosing.

If you prefer not to create a fictional client, you may use the details below:

Example Client

- Name: *HotView Labs*
- Sector: *Tech Research and Development*
- Size: *30 employees*
- Location: *Worldwide*
- Mission: *Provide the most accurate and up to date insights to tech leaders across the world.*

Blog Feed Page

The blog feed page requires the following user stories:

- ☐ As a user, I want to see an interactive banner carousel on the blog feed page, so that I can view a rotation of the 3 latest posts.
- ☐ As a user, I want to click on a button for each carousel item, taking me to the blog post page to read more.
- ☐ As a user, I want to click on the previous or next button in the carousel to animate and reveal another post, to ensure I can see different posts easily.
- ☐ As a user, I want the carousel to return to the first post after reaching the end of the list, and vice versa when clicking previous on the first post.
- ☐ As a user, I want to view a static list of the 12 latest posts in a responsive thumbnail grid on the blog feed page, so I can easily select which post to read.
- ☐ As a user, I want each thumbnail in the blog post feed to be clickable, taking me to the blog post page to read more.

Blog Post Public Page

The blog post public page requires the following user stories:

- ☐ As a user, I want to see a responsive layout showing the post title, author, publication date, image banner, and post content from the API.
- ☐ As a user, I want each blog page to have a shareable URL including a query string or hash parameter that contains the post ID, so I can share the post with others easily.

Blog Post Edit Page

The blog post edit page requires the following user stories:

- ☐ As the owner, I want the blog post edit page to be available only for me when logged in, to ensure no unauthorized edits can be made to my posts.
- ☐ As the owner, I want a delete button that sends a DELETE request to the API for this post ID on the edit page, so I can easily remove my post if needed.
- ☐ As the owner, I want a validated edit form that allows me to update the title, body content, or image by sending a POST request to the API for this post ID, ensuring I can keep my posts up to date easily.

Account Login Page

- ☐ As the owner, I want a validated login form that allows me to request and save a token to my browser by entering my email and password, allowing me to manage posts.

Account Register Page

- ☐ As the owner, I want a validated register form that allows me to create a new account by entering my name, email and password.

Sitemap

The client requires the following pages in their sitemap:

- /index.html (Blog Feed Page)
- /post/index.html (Blog Post Public Page)
- /post/edit.html (Blog Post Edit Page)
- /account/login.html (Account Login Page)
- /account/register.html (Account Register Page)

Please note that these exact pages are required for SEO purposes.

API Platform

You have been provided with a set of API URLs that can be found here:

- [Swagger](#)
- [API Documentation](#)

Key Endpoints

The following API endpoints will be required to complete this project:

- GET /blog/posts/<name>
- GET /blog/posts/<name>/<id>
- PUT /blog/posts/<name>/<id>
- POST /blog/posts/<name>/<id>
- DELETE /blog/posts/<name>/<id>
- POST /auth/register
- POST /auth/login

Please note that <name> will be replaced by your registered username and <id> will be replaced by the ID value of a given blog post item.

Level 1 Process

The following steps are required to pass this examination project:

1. Accept the [GitHub Classroom](#) brief here
2. Clone your Classroom Repository using GitHub Desktop
3. Create a plan for your project using GitHub Projects
4. Create wireframes for your project using Figma
5. Create a style guide and high fidelity assets using Figma
6. Use your plan and design documents to complete the user stories
7. Manually test each user story to ensure it is completed
8. Deploy your project to a static hosting platform
9. Use online testing tools to validate HTML and WCAG
10. Deliver the required files ([see below](#))

Level 2 Process (optional)

The following steps are not required to pass this examination project, however completing one or more of these will distinguish you as a student.

1. Implement pagination to allow for users to view more than 12 posts.
2. Implement a sorting mechanism to allow for users to change the order of posts.
3. Implement a filter mechanism to allow for users to narrow the list of posts.
4. Implement a search mechanism to allow for users to find a specific post.
5. Implement container queries to enable an advanced responsive layout.

Required Deliverables

- ☐ Link to your GitHub classroom repository
- ☐ Link to your public hosted demo
- ☐ Link to your Figma design assets

- ☐ Link to your public planning board
- ☐ Login details for the admin user (email and password)